

PAT-NO: JP411224278A  
DOCUMENT-IDENTIFIER: JP 11224278 A

TITLE: MASK PATTERN CHECKING METHOD

PUBN-DATE: August 17, 1999

INVENTOR-INFORMATION:

NAME	COUNTRY
YAMANO, MASAKAZU	N/A

ASSIGNEE-INFORMATION:

NAME	COUNTRY
RICOH CO LTD	N/A

APPL-NO: JP10027060

APPL-DATE: February 9, 1998

INT-CL (IPC): G06F017/50

ABSTRACT:

PROBLEM TO BE SOLVED: To verify what kind of a mask pattern a final mask pattern uses and to prevent the generation of a mask pattern defect or the like by attaching version numbers to the mask patterns for respective function cells and extracting the respective version numbers of the mask patterns for constituting a semiconductor integrated circuit.

SOLUTION: Whether or not a retrieved cell name is a cell defined in a latest version description file is judged (step S2). A cellversion number is extracted and obtained from the mask pattern of a cell to be checked (step S3). The latest version number of the cell to be checked is obtained from the latest version description file (step 34). The version number obtained from the mask

pattern and the version number obtained from the latest version  
description

file are compared (step S5). Thus, checking is performed by a simple  
method

without performing layout comparison which requires a certain  
processing time.

COPYRIGHT: (C) 1999, JPO

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-224278

(43) 公開日 平成11年(1999) 8月17日

(51) IntCl.<sup>8</sup>  
G 0 6 F 17/50

識別記号

F I  
G 0 6 F 15/60

6 6 6 Z

6 1 4 D

審査請求 未請求 請求項の数4 O L (全 5 頁)

(21) 出願番号 特願平10-27060

(22) 出願日 平成10年(1998) 2月9日

(71) 出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72) 発明者 山野 雅一

東京都大田区中馬込1丁目3番6号 株式会社リコー内

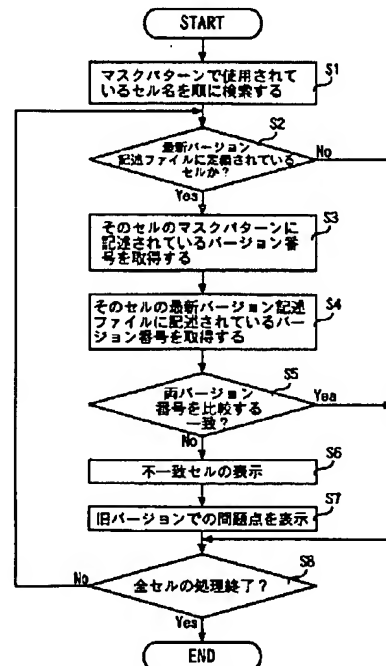
(74) 代理人 弁理士 鳥居 祥

(54) 【発明の名称】 マスクパターンチェック方法

(57) 【要約】

【課題】 この発明は、最終的なマスクパターンがどのようなマスクパターンを用いているのか検証可能にし、マスクパターン不良の発生等を防止することを目的とする。

【解決手段】 この発明のマスクパターンチェック方法は、機能セル毎のマスクパターンにバージョン番号を付けると共に、半導体集積回路を構成するマスクパターンのそれぞれのバージョン番号を抽出し、前記抽出したバージョン番号を対応する機能セルの最新バージョン番号と比較し、最新バージョンのセルのマスクパターンを使用しているかをチェックする。



## 【特許請求の範囲】

【請求項1】 機能セル毎のマスクパターンにバージョン番号を付けると共に、半導体集積回路を構成するマスクパターンのそれぞれのバージョン番号を抽出することを特徴とするマスクパターンチェック方法。

【請求項2】 前記抽出したバージョン番号を対応する機能セルの最新バージョン番号と比較し、最新バージョンのセルのマスクパターンを使用しているかをチェックすることを特徴とする請求項1に記載のマスクパターンチェック方法。

【請求項3】 バージョン番号に不一致があった場合、旧バージョンでの問題点や変更理由を表示することを特徴とする請求項2に記載のマスクパターンチェック方法。

【請求項4】 機能セルであるかの判定をするためのセル種別記述ファイルに記述されているセル毎にマスクパターンをライブラリのマスクパターンと比較すると共に、コンパイルドセルは、使用されているパラメータを元に最新のパターンを再合成してチェックすることを特徴とするマスクパターンチェック方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】この発明は、例えば、大規模集積回路（LSI）開発におけるレイアウトパターンの設計において、作成したマスクパターンの検証方法に関するものである。

## 【0002】

【従来の技術】半導体集積回路のマスクパターンを作成するまでには、回路設計から最終的なマスクパターンを得るまでに数ヶ月あるいはそれ以上に及ぶ期間がかかる。これらの期間、設計者はライブラリから必要なセルのマスクパターンを取り出して使用するが、ライブラリから取り出した後に、ライブラリのマスクパターンが変更されることがある。このライブラリの変更は、通常、マスクパターンの異常を修正する場合や、スペック改良、その他歩止まり向上のため等の理由で行なわれる。

【0003】設計者がこのライブラリの変更を知らずにいると最終的なマスクパターンとライブラリのマスクパターンが不一致となり、その結果、設計した集積回路のマスクパターンに不具合が生じる場合がある。

【0004】また、一旦設計が修了した集積回路を改訂して新たな集積回路を作成する場合もある。この場合には以前に使用したマスクパターンをそのまま一部を再利用する場合にも、改訂する前にライブラリが変更されている場合があり、この場合も同様な問題が発生することがある。

【0005】一方、マスクパターン自体を比較する方法が特開平6-326191号公報に示されている。この公報には、マスクパターン比較を高速に行う技術について開示されているが、比較するための基準マスクパター

ンの変更があった場合の対応については何等対策が講じられていない。

## 【0006】

【発明が解決しようとする課題】この発明は、設計に際してライブラリから取り出した後に、ライブラリのマスクパターンが変更された場合に問題が生じることがないように、最終的なマスクパターンがどのようなマスクパターンを用いているのか検証可能にし、マスクパターン不良の発生等を防止することを目的とする。更に、この発明は、最終のライブラリデータが使用されているかをチェックすることを目的とする。

## 【0007】

【課題を解決するための手段】この発明のマスクパターンチェック方法は、機能セル毎のマスクパターンにバージョン番号を付けると共に、半導体集積回路を構成するマスクパターンのそれぞれのバージョン番号を抽出することを特徴とする。

【0008】前記抽出したバージョン番号を対応する機能セルの最新バージョン番号と比較し、最新バージョンのセルのマスクパターンを使用しているかをチェックするように構成すると良い。

【0009】上記した構成によれば、処理時間のかかるレイアウト比較を行わずに簡易な方法で最新のライブラリデータを使用しているかをチェックできるため、高速にチェックができる。

【0010】更に、バージョン番号に不一致があった場合、旧バージョンでの問題点や変更理由を表示するように構成すると良い。

【0011】上記のように構成することで、旧バージョンのマスクパターンを使用していた場合には、旧バージョンでの問題点や変更理由を表示することができるため、そのまま旧バージョンを使用した場合にどのような問題が発生するかを知ることができる。また、その理由によって、問題のあるセルのマスクパターンを入れ換えるべきかの判定を行うことができる。

【0012】また、この発明のマスクパターンチェック方法は、機能セルであるかの判定をするためのセル種別記述ファイルに記述されているセル毎にマスクパターンをライブラリのマスクパターンと比較すると共に、コンパイルドセルは、使用されているパラメータを元に最新のパターンを再合成してチェックすることを特徴とする。

【0013】上記の構成によれば、マスクパターン全体のマスクパターン比較を行わず、機能セル単位でチェックを行うものであるが、ライブラリにマスクパターンデータとして存在しないコンパイルドセルについても最新のものが使用されているかのチェックが可能となっている。このチェック方法では、バージョン番号を比較する前述した構成に比べて、処理時間がかかるが、バージョン情報の管理が不要となると共に、さらにマスクパタ

ーン自体を比較するため、確実にチェックが行える。

【0014】

【発明の実施の形態】以下、この発明の実施の形態につき図面を参照して説明する。図1は、この発明が適用されるマスクパターンチェック装置の基本的構成を示すブロック図である。

【0015】図1に示すように、処理装置1は、CPU及びこれら動作を制御するプログラムが格納された記憶装置を備え、後述する動作フローに従ってマスクパターンをチェックする処理を行う。この処理装置1には、キーボード及びマウスなどのポインティングデバイスを含む入力装置2からチェックの際のコマンドが入力される。更に、この処理装置1には、マスクパターンデータやセル情報を記述ファイルとして格納している記憶装置3からそれぞれのデータが与えられ、マスクパターンのチェックを行う。このチェック結果はCRTなどからなる表示装置4に表示される。

【0016】次に、上記処理装置の動作につき動作フロー図に従い説明する。この第1の実施の形態においては、必要な機能セル毎にバージョン番号がそれぞれ付加された状態でライブラリとして記憶装置3に格納されている。機能セルとしては、論理ゲート、メモリブロック、マルチプレイア等があり、図4に示すように、マスクパターンにバージョン番号が付され、ライブラリが改訂された毎にこのバージョン番号は書き換えられる。そして、図5に示すように、記憶装置3内に最新バージョン記述ファイルとして格納されている。設計者が設計したマスクパターンに使用した時点のライブラリのバージョン番号がそのマスクパターンのライブラリと共に記述ファイルとして格納されている。そして、マスクパターンをチェックする場合には、検証するマスクパターンの記述ファイル等を記憶装置3内に格納する。

【0017】まず、この発明の第1の実施の形態につき図2の動作を示すフローに従い説明する。

【0018】まず、チェック動作を開始すると、ステップS1において、記憶装置3に格納されているチェックするマスクパターンの記述ファイルからマスクパターンで使用されているセル名を順次検索する。

【0019】続いて、ステップS2において、検索したセル名が最新バージョン記述ファイルに定義されているセルかどうかを判断する。定義されているセルの場合は、バージョンのチェックを行うためにステップS3に進み、定義されていない場合は、ステップS8の処理を行う。

【0020】最新バージョン記述ファイルの記述例を図5に示す。この例は最新バージョン記述ファイルのフォーマットを限定するものではないが、内容としては、機能セルのセル名とその最新バージョンが記述されているものである。このステップS2における判断は、マスクパターンで使用されているセルには機能セル以外のバー

ジョンのチェックが不要なものも含まれており、このセルを除くことで余分なチェックを省き、高速に処理を行うためである。チェックが不要なセルの一例として、機能セルを構成する下階層のセルが挙げられる。ただし、これらの下階層セルもチェックに含まれるのであれば、必要なものをチェックするようにしても良い。

【0021】ステップS3においては、チェックを行うセルのマスクパターンからそのセルのバージョン番号を抽出して取得し、ステップS4に進む。図4にバージョン番号が記述されているマスクパターンの例を示す。この例では、“CELL\_A”というセルのマスクパターンのバージョンが“3”であることを示している。

【0022】ステップS4においては、チェックを行うセルの最新バージョン番号を最新バージョン記述ファイルから取得する。図5の例では、“CELL\_A”のバージョン番号“3”が得られる。

【0023】ステップS5においては、マスクパターンから得たバージョン番号と、最新バージョン記述ファイルから得たバージョン番号を比較する。一致する場合は、ステップS8に進み、一致しない場合にはステップS6に進み、処理を行う。図4及び図5に示す例では、“CELL\_A”のバージョン番号が各々3であるため一致している。

【0024】セルが一致しない場合、例えば、マスクパターンから得た“CELL\_A”のバージョン番号が1であり、最新バージョン記述ファイルから得た“CELL\_A”のバージョン番号が3である場合には、ステップS5において、両者のバージョン番号が一致しないと判断され、ステップS6にて、バージョン番号の一致しないセル情報の表示を行った後、ステップS7に進む。この例では、図6の一行目に示したように、「CELL\_Aのバージョンは1ですが、最新バージョンは3です。」という表示を表示装置4に行う。尚、この表示は例であって限定するものではない。このようにバージョン番号を比較することにより、処理時間のかかるレイアウト比較を行わずに簡易な方法で最新のライブラリデータを使用しているかをチェックできるため、高速にチェックができる。

【0025】続いて、ステップS7では、ここでは、旧バージョンを使用した際の問題点を表示する。問題点とは、マスクパターンで使用されていた旧バージョンから最新バージョンまでの改定履歴で、これらを各バージョンについて表示する。このように、改訂履歴を表示することで、そのまま旧バージョンを使用した場合にどのような問題が発生するかを知ることができる。また、その理由によって、問題のあるセルのマスクパターンを入れ換えるべきかの判定を行うことができる。

【0026】ステップS8では、マスクパターンで使用されているすべてのセルについて、ステップS2以降の処理がなされたかを判定する。すべてのセルについて終

了した場合は、処理を終了し、残りのセルがある場合は、ステップS2へ戻り前述の動作を繰り返す。

【0027】次に、この発明の第2の実施の形態につき図3のフロー図に従い説明する。単純にマスクパターン比較を行うためのツールは、多くのEDAベンダーから販売されている。この発明の第2の実施の形態では、これらのマスクパターン比較のツール等を用いて、マスクパターンの最新バージョンとの比較を行う。この第2の実施の形態においては、機能セル毎にマスクパターン比較を行っている。

【0028】マスクパターンのチェックを開始すると、まず、ステップS11において、セル種別記述ファイルに記述されているセルについて順に検索する。セル種別記述ファイルの例を図7に示す。このファイルにはライブラリに用意されているセルとコンパイルドセルのパラメータが記述されている。この例では、記述されている3つのセルについて、“CELL\_A”、“CELL\_B”はライブラリで用意されているセルで、“CELL\_C”は、ビット数が8、ワード数が1024のRAMのコンパイルドセルであることを示している。

【0029】続いて、ステップS12において、ステップS11で得たセルが最新バージョンに定義されているマスクパターンで使用されているセルであるかを判断する。該当しない場合は、ステップS18に進み、ステップS18の処理を行う。ステップS12において、該当する場合は、ステップS13に進み、ステップS13でコンパイルドセルであるか否かの判断を行う。

【0030】ステップS13において、マスクパターンに使用されているセルがコンパイルドセルでない場合は、ステップS15にそのまま進む。

【0031】ステップS13において、セルがコンパイルドセルであると判断されると、ステップS14に進み、ステップS14では、コンパイルドセルについてコンパイルドセルのパラメータを元にそのセルの最新バージョンのマスクパターンを発生する。これは、コンパイルドセルは、ライブラリとして完成された形で存在せず、コンパイルドセルを構成するサブセルを組み合わせて構成されるからである。図7に示す例では、ビット数が8、ワード数が1024のRAMの最新バージョンのマスクパターンを発生することになる。

【0032】ステップS15では、ライブラリ及び/または、ステップS14において発生させたコンパイルドセルの最新バージョンのマスクパターンの比較を行う。ここでの比較は図形的な比較であり、この比較の手法としては、EDAベンダーから販売されているツール等や特開平6-326191号公報に開示されているような圧縮したビットマップに基づく比較処理などを用いることができる。

【0033】続いて、ステップS16では、ステップS15における比較の結果に基づきマスクパターンが一致

するか否かを判断し、一致するマスクパターンであれば、ステップS18に進み、一致しない場合は、ステップS17に進む。ステップS17で、前述した第1の実施の形態と同様に不一致セルの情報を表示装置4に表示する。

【0034】上記したように、マスクパターン全体のマスクパターン比較を行わず、機能セル単位でチェックを行うものであるが、ライブラリにマスクパターンデータとして存在しないコンパイルドセルについても最新のものが使用されているかのチェックが可能となる。このチェック方法では、第1の実施の形態に比べて、処理時間がかかるが、バージョン情報の管理が不要になる点、さらにマスクパターン自体を比較するため、確実にチェックが行えることなどの利点がある。

【0035】ステップS18では、セル種別記述ファイルに記述されているセル全てについて、上記ステップS12以降の処理がなされたかを判定する。全てのセルについて終了した場合は、処理を終了し、残りのセルがある場合は、ステップS12に戻り前述した処理を行う。

【0036】

【発明の効果】以上説明したように、この発明の請求項1及び2に記載のマスクパターンチェック方法においては、処理時間のかかるレイアウト比較を行わずに簡易な方法で最新のライブラリデータを使用しているかをチェックできるため、高速にチェックができる。

【0037】また、この発明の請求項3に記載のマスクパターンチェック方法においては、旧バージョンのマスクパターンを使用していた場合には、旧バージョンでの問題点や変更理由を表示することができるため、そのまま旧バージョンを使用した場合にどのような問題が発生するかを知ることができる。また、その理由によって、問題のあるセルのマスクパターンを入れ換えるべきかの判定を行うことができる。

【0038】また、この発明の請求項4に記載のマスクパターンチェック方法においては、マスクパターン全体のマスクパターン比較を行わず、機能セル単位でチェックを行うものであるが、ライブラリにマスクパターンデータとして存在しないコンパイルドセルについても最新のものが使用されているかのチェックが可能となっている。このチェック方法では、請求項1及び2に記載の場合に比べて、処理時間がかかるが、バージョン情報の管理が不要になること、さらにマスクパターン自体を比較するため、確実にチェックが行えることなどの利点がある。

【図面の簡単な説明】

【図1】この発明が適用されるマスクパターンチェック装置の基本的構成を示すブロック図である。

【図2】この発明の第1の実施の形態の処理動作を示すフロー図である。

【図3】この発明の第2の実施の形態の処理動作を示す

フロー図である。

【図4】バージョン番号が記述されているマスクパターンの例を示す模式図である。

【図5】最新バージョン記述ファイルの記述例を示す模式図である。

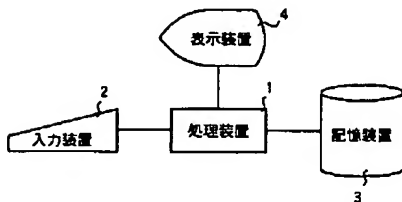
【図6】旧バージョンでの問題点の表示例を示す模式図である。

【図7】セル種別記述ファイルの例を示す模式図である。

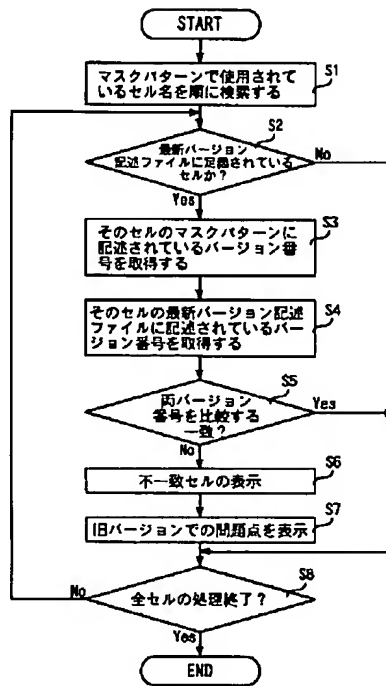
【符号の説明】

- 1 処理装置
- 2 入力装置
- 3 記憶装置
- 4 表示装置

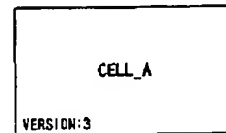
【図1】



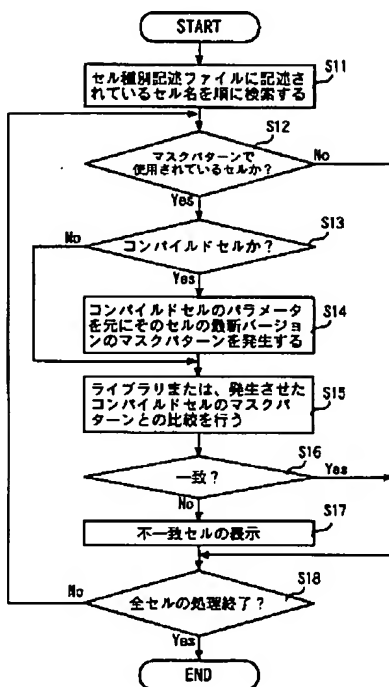
【図2】



【図4】



【図3】



【図5】

CELL_A	3
CELL_B	1
CELL_C	2
.	.

【図7】

CELL_A	L1B
CELL_B	L1B
CELL_C	COMPILED RAM BIT:8 WORD:1024
.	.

【図6】

CELL\_Aのバージョンは1ですが、最新バージョンは3です。  
CELL\_Aの改定履歴  
バージョン2では、歩止まり向上の処置が行われています。  
バージョン3では、出力端子OUTのドライブ能力が2倍に向上されました。